Experiment 5

Student Name : Misal Saini
UID: 23BCS13467
Branch: CSE
Section/Group: KRG_2B
Semester: 5th
Date of Performance: 25/09/25
Subject Name: PBLJ
Subject Code: 23CSH-304

1. Aim:

To design and implement Java programs using autoboxing, serialization, file handling, and efficient data processing and management.
• To apply Wrapper classes, object serialization, and Java I/O concepts in solving real-world problems.

- **Part A – Easy Level:**

• To create a Java program that calculates the sum of a list of integers using autoboxing and unboxing.
• To parse strings into wrapper objects and demonstrate automatic conversion between primitives and objects.

- **Part B – Medium Level:**

• To create a Java program that serializes and deserializes a `Student` object using Java I/O streams.
• To handle exceptions like `FileNotFoundException`, `IOException`, and `ClassNotFoundException` during file operations.

- **Part C – Hard Level:**

• To create a menu-based Java program for storing and displaying employee details using file handling.
• To implement console-driven interaction for adding, displaying, and managing employee records in a file.

2. Objective:

- ✓ To understand the concept of autoboxing and unboxing using Java Wrapper classes.
- ✓ To implement object serialization and deserialization for persistent storage of objects.
- ✓ To practice Java file handling using readers, writers, and object streams.
- ✓ To apply exception handling for robust execution of file I/O operations.

3. JAVA script and output:

EASY-LEVEL PROBLEM

```
package exp.pkg5;

import java.util.*;

public class Exp5 {
    public static void main(String[] args) {
        Scanner sc = new Scanner(System.in);
        System.out.print("Enter numbers (comma separated): ");
        String input = sc.nextLine();
        String[] arr = input.split(",");
        ArrayList<Integer> list = new ArrayList<>();
        for (String s : arr) {
            list.add(Integer.parseInt(s.trim()));
        }
        int sum = 0;
        for (int num : list) {
            sum += num;
        }
        System.out.println("Sum of numbers = " + sum);
    }
}
```

OUTPUT:

```
run:
Enter numbers (comma separated): 22,45,67,9
Sum of numbers = 143
BUILD SUCCESSFUL (total time: 10 seconds)
```

Figure 1: Easy Level

MEDIUM LEVEL PROBLEM:

```
package exp.pkg5;
import java.io.*;
class Student implements Serializable {
    int id;
    String name;
    double gpa;
    Student(int id, String name, double gpa) {
        this.id = id;
        this.name = name;
        this.gpa = gpa;
    }
}

public class Exp5 {
    public static void main(String[] args) {
        try {
            Student s1 = new Student(101, "Akshara", 9.1);
```

```
        ObjectOutputStream oos = new ObjectOutputStream(new
FileOutputStream("student.dat"));
        oos.writeObject(s1);
        oos.close();
        System.out.println("Student serialized successfully!");
        ObjectInputStream ois = new ObjectInputStream(new FileInputStream("student.dat"));
        Student s2 = (Student) ois.readObject();
        ois.close();
        System.out.println("Student deserialized:");
        System.out.println("ID: " + s2.id);
        System.out.println("Name: " + s2.name);
        System.out.println("GPA: " + s2.gpa);
    } catch (FileNotFoundException e) {
        System.out.println("File not found.");
    } catch (IOException e) {
        System.out.println("IO Exception: " + e.getMessage());
    } catch (ClassNotFoundException e) {
        System.out.println("Class not found.");
    }
  }
}
```

OUTPUT:

```
Student serialized successfully!
Student deserialized:
ID: 101
Name: Akshara
GPA: 9.1
BUILD SUCCESSFUL (total time: 0 seconds)
```

Figure 2: Medium Level

HARD LEVEL PROBLEM

```
package exp.pkg5;

import java.io.*;
import java.util.*;

public class Exp5 {
    public static void main(String[] args) {
        Scanner sc = new Scanner(System.in);
        while (true) {
            System.out.println("Menu:\n1. Add Employee\n2. Display All\n3. Exit");
            System.out.print("Enter choice: ");
            int choice = sc.nextInt();
            sc.nextLine();
            if (choice == 1) {
                try {
                    BufferedWriter bw = new BufferedWriter(new FileWriter("employees.txt", true));
                    System.out.print("Name: ");
                    String name = sc.nextLine();
                    System.out.print("ID: ");
                    int id = sc.nextInt();
                    sc.nextLine();
                    System.out.print("Designation: ");
```

```
                String desig = sc.nextLine();
                System.out.print("Salary: ");
                double sal = sc.nextDouble();
                sc.nextLine();
                bw.write(name + "," + id + "," + desig + "," + sal);
                bw.newLine();
                bw.close();
                System.out.println("Employee added successfully!");
            } catch (IOException e) {
                System.out.println("Error writing to file.");
            }
        } else if (choice == 2) {
            try {
                BufferedReader br = new BufferedReader(new FileReader("employees.txt"));
                String line;
                System.out.println("Employee Records:");
                while ((line = br.readLine()) != null) {
                    String[] data = line.split(",");
                    System.out.println("Name: " + data[0] + ", ID: " + data[1] + ", Designation: " +
data[2] + ", Salary: " + data[3]);
                }
                br.close();
            } catch (IOException e) {
                System.out.println("Error reading file.");
            }
        } else if (choice == 3) {
            System.out.println("Exiting...");
```

```
            break;
        } else {
            System.out.println("Invalid choice.");
        }
    }
}
```

OUTPUT:

```
Menu:
1. Add Employee
2. Display All
3. Exit
Enter choice: 1
Name: Akshara
ID: 11410
Designation: Software Engineer
Salary: 39000
Employee added successfully!
Menu:
1. Add Employee
2. Display All
3. Exit
Enter choice: 2
Employee Records:
Name: Akshara, ID: 11410, Designation: Software Engineer, Salary: 39000.0
Menu:
1. Add Employee
2. Display All
3. Exit
Enter choice: 3
Exiting...
BUILD SUCCESSFUL (total time: 30 seconds)
```